

# IT Governance Mechanisms for DevOps Teams – How Incumbent Companies Achieve Competitive Advantages

Anna Wiedemann

Neu-Ulm University of Applied Sciences  
Center for Research on Service Sciences (CROSS)  
[anna.wiedemann@hs-neu-ulm.de](mailto:anna.wiedemann@hs-neu-ulm.de)

## Abstract

*More and more organizations are deciding to move from traditional, plan-driven software development to agile approaches in order to stay competitive. Therefore, the IT functions have been deciding to implement cross-functional DevOps teams. To enable collaboration within DevOps teams, incumbent companies have to implement mechanisms to govern dynamic and agile environments. The present research investigates which IT governance mechanisms are helpful for the implementation of DevOps teams. For this purpose, we conducted a qualitative research study and interviewed team members in six companies that have already implemented DevOps-oriented teams. We describe which IT governance mechanisms—in the form of structure, processes, and relational mechanisms—are important for DevOps teams to achieve competitive advantages. Our findings show that agile roles and responsibilities, hybrid or decentralized organizational structures, as well as communications and knowledge-sharing models are conducive to the government of a DevOps team.*

## 1. Introduction

Established organizations are under pressure to speed up with the rapid innovation management provided by start-up companies [1]. Many traditional IT functions are not able to react fast to customers' needs with their current set-ups. Often, the IT functions are very formally organized, with little flexibility. They have to implement structures, processes and abilities that suit today's customers' requirements [2]. Within the digital age, the agility of IT functions has become a key factor in driving innovation and gaining competitive advantage. Now, more and more IT organizations recognize that they have to shift from the traditional service-provider role

to more agile oriented approaches to become a partner for the business [3-5]. Agility is defined as “the ability to respond operationally and strategically to changes in the external environment. The response has to be quick and effective for the organization to be considered agile” [6], p. 444.

One of the most important differences between organizations that follow agile approaches and organizations that follow more traditional approaches is that the agile ones establish autonomous, self-organized teams. In this way, learning beyond knowledge silos is facilitated, making enhanced and advanced decisions-making possible within these teams [7].

To stay competitive, established companies have to adapt their internal organizations and their forms of collaboration as well as their alignment with the business [2]. By establishing agile methods and new technologies it is possible to develop fast software features and deploy them for a rapid response to customer requirements. But agile methods are often only used by software developers [8], but for the delivery of fast and new software features, other activities of the IT departments are necessary, too. Striving for the creation and sustainability of digital innovations, incumbent companies try to implement blueprints of start-ups within their IT departments [9]. In particular, they increasingly copy the approaches of successful start-ups like Spotify [10, 11] and aim to establish tight collaboration between product managers, developers and operating staff within their existing organizational structures [11, 12]. For example, Amazon and Spotify replaced their traditional “silo” functions, which required enormous coordination efforts, with autonomous, cross-functional, product-centric teams that include a maximum of eight people [8]. This enables them to gain, share and implement knowledge, speed up decision-making processes significantly and thus, to meet demand in rapid changing environments [13]. These organizational units often follow the DevOps approach. DevOps is a clipped compound of the

words development and operation. The process it refers to entails strong collaboration between development and operations, automation, and use of new tools and technologies [14, 15]. One of the major differences between traditional IT departments and cross-functional DevOps teams is that the latter need to be governed by lightweight models [9].

The implementation of the DevOps approach causes changes in the internal IT functions. These changes are reflected in new structures, processes and other governance mechanisms. Some organizations have already started to adapt their IT functions. Incumbent companies have to rethink their IT governance mechanisms within dynamic and agile environments [16]. However, in Information Systems (IS) research, it is still not clear how incumbent firms could be enabled to build and implement such IT governance mechanisms to achieve competitive advantage. Hence, we posed the following research question: *How can incumbent firms implement suitable IT governance mechanisms for DevOps teams to achieve competitive advantage?*

The remainder of the paper is structured as follows. In the next section the theoretical foundation of this study is outlined, in which we review the concept of DevOps and relate these to IT governance mechanisms. Afterwards, we outline the case study approach adopted here. Subsequent to the presentation of the findings, we discuss how our findings extend existing research and conclude with theoretical and practical implications.

## 2. Theoretical background

### 2.1 The DevOps concept

During the last years, the new phenomenon of DevOps appeared to manage the development and the corresponding deployments of new software. The DevOps approach is a new phenomenon in software engineering. The aim of DevOps is to enhance collaboration, automation, virtualization as well as tools to bridge activities of software development and operation [15, 17]. Software development activities include requirements management, software design, implementation, test management and integration; whereas software operation focusses on maintenance and software installation tasks. Through DevOps, solutions are delivered to avoid interruptions between different stages of the complete software delivery process [18]. The entire software development life-cycle contains the steps of planning, developing and operation tasks. Through DevOps, companies are enabled to frequently and automatic release new

software features. Hence, risks that are linked with software releases can be reduced, and feedback of new software features is received faster [15]. In addition, agile software development methods can be used to manage software development [15].

For the fast delivery of new software features, innovations and quick handling of problems, IT departments should implement cross-functional teams rather than separated silo architectures. All necessary activities for the software delivery cycle of one service should be conducted by a single team. DevOps broadens the agile approach, e.g. by applying continuous integration. Continuous integration is defined as a process that is provoked automatically and encompasses interconnected stages, e.g. an acceptance test, code validation, a compliance check as well as release package development [18]. Additionally, it helps to prevent interruptions between the development and the deployment stages of software delivery. Furthermore, it enhances the release process through better collaboration between developers and operation employees, and delivers fast value for business [12].

### 2.2 IT governance

Prior literature highlighted the idea that companies that have distinct IT governance models do better than their competitors [19, 20]. But as mentioned before, more and more IT functions implement cross-functional and agile IT teams [3, 5]. As previous research has pointed out, managerial issues are very challenging if IT organizations adopt agile methods. The implementation of cross-functional, autonomous teams requires a corresponding IT governance [21, 22]. IT governance is a novel mode of IT management. IT management focuses on providing IT operations and services. Organizations IT governance is specific and cannot simply delegate to an external partner. IT governance on the other hand focuses on internal and external environments. IT governance helps to meet the needs of the internal business departments and the IT functions can be transformed to react to any new and unexpected demands made by external business customers [23, 24]. The IT governance mechanisms that are useful for traditional organized IT functions need to be adapted toward a lightweight IT governance that gives teams the necessary guidelines [9]. The aim of IT governance is to achieve strategic business/IT alignment [25]. Since the Strategic Alignment Model was introduced by Henderson and Venkatraman [26], a lot of research has been presented describing that approach. But alignment is required not only at the strategic level, but also at the

operational level [24]. Hence, IT governance is also applicable to a more operational area, e.g. for coordinating day-to-day operations [22].

Van Grembergen [24] pointed out that structure, processes, and relational mechanisms are the key elements of IT governance. *Structure* comprises defined roles and responsibilities, as well as committees for each division of the company. *Processes* consist of decision-making, as well as planning and monitoring such that IT policies are suitable to business needs. The third element, *relational mechanisms*, refers to the exchange between IT and business, dialogs, shared knowledge, and communication [25, 27].

Another IT governance approach was presented by Weill and Ross [19]. They highlighted the idea that IT governance exists of three mechanisms, namely: decision-making structures, alignment processes, and communication approaches. The *decision-making structures* are the organizational units such as committees, executives' meetings or other business/IT executives' agreements for decision-making. The *alignment processes* consist of IT's alignment with organizational policies through the definition of formal processes such as IT investments and evaluation. *Communication approaches* allow a better communication of principles and policies of IT governance and decisions within the enterprise [19, 27].

This paper describes how an IT department can create an IT governance system to suit the implementation of DevOps teams. The IT governance concepts of Van Grembergen [24] and Weill and Ross [19] are very similar [27]. For our investigation we adapt the structure and processes from both studies and the relational mechanisms from De Haes and Van Grembergen [28]. Within DevOps teams, communication and knowledge sharing play a significant role [9]. Hence, we decided to include the governance mechanism of relational mechanisms. The ideal setting for IT governance mechanisms will be different in every environment and is dependent on several eventualities [25]. Furthermore, looking at DevOps environments, a lightweight IT governance within an agile setting [9] and the implementation of IT governance practices in daily operation is needed [29].

To enhance the IT functions of alignment and scaling agility, team-oriented collaboration models such as the Spotify framework are used [30]. This framework emphasizes feature-based autonomous teams, called "squads". Squads that work in similar areas are combined into "tribes". Within these tribes, employees with a similar background form "chapters". These chapters help people to share

knowledge and improve their understanding. Communities of interests are bundled into "guilds". Guilds foster the discussion of practices and knowledge. Additionally, they enable communication within organizations across tribes [9, 30].

### 3. Research design

In part to enhance our understanding of how incumbent firms can implement suitable governance mechanisms for DevOps, an exploratory multiple-case study approach was adopted. The case study approach is defined as "*an empirical inquiry that investigates a contemporary phenomenon in depth and within its real-life context*" [31], p. 18. The case study method is suitable for exploratory studies, to identify relevant constructs. Furthermore, for the present study, expert knowledge about a contemporary, complex approach was necessary (the how and why of the context). The case study approach is suitable for complex and unexplored organizational processes, e.g. organizational changes or software implementation projects [32]. This is one of the first studies that investigates IT governance mechanisms for DevOps teams [31]. The advantage of case study research is that it can zoom in on real-life situations and test or develop theoretical perspectives in relation to phenomena as they unfold in practice [33]. Hence, summarizing case studies is an appropriate method to improve our understanding of the IT governance mechanisms structure, processes, and relational mechanisms of DevOps teams. The units of analysis are DevOps team members and their managers.

The case studies approach can follow different research designs such as single-case and multiple-case designs. To improve our insights into the phenomenon and to increase the validity of the findings, we decided to conduct a multiple-case study [31]. Our case selection aimed at capturing the range of variation in a subset of units in which the mechanisms of interest can be observed [34]. We will outline the case selection method in greater detail in the next chapter.

#### 3.1 Case selection

To identify and select appropriate cases several criteria were set. First, the team must have implemented the DevOps principles at least six months. Second, the team must be part of an incumbent firm. Third, at least one senior manager (e.g., the team leader) and a person concerned with operational tasks need to consent to being

**Table 1. Characteristics of case study participants**

	<b>Team 1</b>	<b>Team 2</b>	<b>Team 3</b>	<b>Team 4</b>	<b>Team 5</b>	<b>Team 6</b>
<b>Industry</b>	Media	Consumer Portal	Pet Supplies	Furniture	Bank	Media
<b>Interviewee</b>	Executive (1.1), Executive (1.2)	Executive (2.1), Senior Engineer (2.2)	Team Lead (3.1), Senior Engineer (3.2)	CTO (4.1), Executive (4.2)	Executive (5.1), Senior Engineer (5.2)	CTO (6.1), Senior Engineer (6.2)
<b>Employees</b>	50-100	500-1.000	100-500	1.000-2.000	2.000-5.000	1-50
<b>Agile method</b>	Scrum	Kanban	Scrum	Kanban	Scrum	Kanban
<b>Digital product/service</b>	Data service	Service for website delivery	Online shop	Online shop	Big data service	Online platform service
<b>DevOps orientation since</b>	Six months	Five years	One year	Two years	Six months	Two to three years

interviewed. Selecting both managers and operational team members should enable us to gain knowledge of the IT governance mechanisms and their effects from a leadership and an operational perspective. To identify cases that comply with these criteria, we browsed through the internet and social networks (e.g. LinkedIn) for business contacts of companies that are engaged in DevOps. We looked for people who had experience with DevOps. For the case studies, over 40 teams from different firms within different industries were contacted via e-mail and telephone. Six companies agreed to participate in the case study. Table 1 depicts the important characteristics of these teams. We applied a multiple-case design that enables cross-case pattern search. This method facilitates the investigation of processes and outcomes over several cases to understand how similar or contrasting results are delivered [31, 34, 35]. For instance, the size of the teams examined varies between four and 23 people. This difference in size led us to assume that the teams had implemented different governance mechanisms.

### 3.2 Data collection and analysis

The data collection phase took place within seven months, from October 2016 through April 2017. A semi-structured interview was conducted with each participant, supported by a guideline that contained a list of questions or general topics that the interviewers wanted to touch on [31, 35]. The questions were mainly open ended, so that the

interviewees had the opportunity to explore their experiences and views [36].

The interview guidelines not only helped to keep the interaction focused as data collection proceeded, it was also used to ensure the comparability of data across individuals, settings, and researchers [37]. Despite making the interviewing process more systematic and comprehensive, the guidelines allowed the interviewer a high degree of freedom to adapt the guidelines to the given situation or interviewee. Thus, questions were adjusted during the interviews to gain more in-depth knowledge on individual cases.

Each interview lasted about 45-75 minutes and was conducted through face-to-face meetings or by telephone. The interviews were held in German or English. German statements were translated into English for further analysis. Every interview was recorded and transcribed. Moreover, a comprehensive number of notes was taken during the interview.

The interview data was coded using the NVivo 10 software application [38]. We started the coding process following the guidelines of Miles and Huberman [35]. Hence, we started with an open coding process. During the coding, notes were taken to justify the coding section. Afterwards, the results were analyzed for the three IT governance mechanisms (structure, processes, and relational mechanisms). Then the findings were compared for each dimension to identify commonalities, relationships, and patterns. The focus was on the constructs that we identified from the literature and the new capabilities that emerged during the data

analysis. Furthermore, we asked the participants of the case study what advantages they experienced after introducing the DevOps approach in their company to gain competitive advantages.

The data was analyzed using within-case analysis as well as cross-case analysis [31]. During the within-case analysis, the codes pertaining to the governance structure, governance processes, and relational mechanisms of the teams were studied and analyzed and every case was seen as standalone entity. With the help of the cross-case analysis, we were able to compare the cases and identify how each governance mechanism is implemented in the cases and foster competitive advantage. The identified constructs

were related to the DevOps concepts and show how DevOps teams govern through the three mechanisms. Finally, we present our findings with the help of a conceptual matrix for each mechanism below [35].

## 4. Findings

Our findings present evidence for the IT governance mechanisms that determine how DevOps teams are organized and how the decision-making authority is implemented within an IT function. For the implementation of DevOps principles, the cases need new technics and technologies. Some cases have invested great efforts into dividing existing

**Table 2. Findings for governance structures**

	<b>Team roles</b>	<b>Decision-making</b>	<b>Organizational structure</b>
<b>Team 1</b>	The team consists mainly of software engineers; one product owner is within the team. Team members are attributed roles, but everyone must be able to take over all the tasks of the software delivery lifecycle. Every engineer must perform the rotating role of operations duty manager.	The team has great autonomy regarding the decision-making process. Every team has a “head” whose function is to act as disciplinarian and coach.	Within the company, all teams are in transition to become DevOps oriented.
<b>Team 2</b>	The team consists mainly of software developers. The team must be able to take over all tasks of the software delivery lifecycle. They work with several product owners from several business sections.	The team has great autonomy in the decision-making process. The managing directors is the disciplinarian for some IT parts.	The company has implemented traditional silo-oriented departments and DevOps-oriented teams.
<b>Team 3</b>	The team consists mainly of software developers, one QA engineer, and one product owner. The team must be able to take over all tasks of the software delivery lifecycle.	The team has great autonomy regarding the decision-making process. A deliveries manager is the disciplinarian and coach.	Within the company, all teams are in transition toward DevOps orientation.
<b>Team 4</b>	The team consists mainly of software developers and one product owner. The team must be able to take over all tasks of the software delivery lifecycle.	The team has great autonomy in the decision-making process. A team lead is the disciplinarian.	The company has implemented mainly DevOps-oriented teams but some services are traditionally organized.
<b>Team 5</b>	Most members of the team are software engineers; one product owner is included in the team. Team members are still attributed roles and are not immediately able to overtake all other team roles.	The team has a certain degree of autonomy. They have distinct processes for each role. A team lead is responsible for the team members.	The company has mainly implemented traditional silo departments and has only a few DevOps oriented teams.
<b>Team 6</b>	Most members of the team are software engineers with one product owner. Team members have fixed roles, but everyone must be able to take over all tasks of the software delivery lifecycle.	The team has great autonomy in the decision-making process. An executive is the disciplinarian for every team.	Within the company, all teams are DevOps oriented.

**Table 3. Findings for governance processes**

Team	Agile method	Core processes
1	Scrum	Requirements Management
2	Kanban	Software Development
3	Scrum	Quality Assurance
4	Kanban	Test Management
5	Scrum	Software Operation
6	Kanban	Support Processes
		Continuous Integration / Delivery / Deployment
		Service Level Agreements

software monoliths into smaller architectural parts (e.g. micro services). In some companies, great efforts were necessary to gain acceptance for the new culture of collaboration within cross-functional teams. In this section, we present tables with condensed cross-case matrixes with our findings for IT governance mechanisms that are integrated in the interviewed companies. Furthermore, we asked the interviewees to identify the major motivation and advantages for the integration of cross-functional teams in their companies. The following tables present our findings for the IT governance mechanisms within the several DevOps teams.

Implementing cross-functional DevOps teams presents challenges for structure. Our findings present evidence that the integration of DevOps takes place in different ways. In Cases 1, 3, 4, and 6 the IT functions majority is organized with cross-functional teams. Case 4 has organized their IT function mainly with DevOps teams, but they still have some traditional organized teams *“for example in the ERP area, where the DevOps oriented approach works rather less (4.1).”* In Cases 2 and 5 most of the IT function uses the traditional silo orientation, but they are starting to implement projects or prototypes of DevOps teams. Furthermore, regarding decision-making structures, all participants of the study mentioned that they have great autonomy within the team. That means that the teams have responsibility for the entire application.

Every team of our study is coordinated by a team lead or directly by the CIO. The cases that have a great DevOps orientation mentioned that management positions are reduced within the company since flat hierarchies are necessary for agility. Hence, within DevOps-oriented structures, *“traditional management tasks break away (1.2)”* and companies have a *“social responsibility towards the former managers (1.2)”* to find suitable positions

for them. In addition, the executives of Case 1 and the CTO of Case 4 mentioned that there is a need for an agile coach for the teams, for example by integrating the Scrum master role. A Scrum master supports the teams to *“avoid inefficiencies and foster homogeneity within the IT function (4.1).”* Regarding the roles and responsibilities, the teams consist mostly of software developers and one product owner. Additionally, team members should be able to take over all the tasks involved in the software delivery cycle. The teams reflect a mix of specialized knowledge and skills *“T-shaped skills [...] team members with profound knowledge in one area,”* CTO case 8 and cross-disciplinary knowledge. In sum, the teams we investigated mentioned that the aim is to take over all the tasks of the software delivery cycle. In some investigated teams, they are already in that working mode.

Table 3 presents the main governance processes which were mentioned by every participant. To gain competitive advantages, the DevOps teams we examined need a couple of core governance processes: Requirements Management, Software Development, Quality Assurance, Test Management, Software Operation, Support, Continuous Integration / Delivery / Deployment, and forms of Service Level Agreements. Furthermore, the teams of Cases 1, 3, and 5 mentioned that they use Scrum as an agile methodology, and that they have implemented the corresponding processes in their daily work. Whereas the teams of Cases 2, 4, and 6 have adopted Kanban for their mode of operation.

To enhance alignment and foster scaling agility within the IT function, organizations use a collaboration and communication framework such as the Spotify model mentioned above. Table 4 depicts our findings for relational mechanisms. Since there is great decision-making autonomy within DevOps teams, the members have to discuss and share their knowledge within the team and the company [9]. Since DevOps teams work very autonomously and have great freedom to make their own decisions, distinct relational mechanisms are necessary. All interviewees mentioned that they have a product owner for their application. Only in Case 2 was the product owner on the business side. The remaining teams had integrated a business person or a person with pronounced knowledge of business processes. *“The role of the product owner [...] is to interact with the business stakeholders (3.2).”* Regarding communication and shared knowledge, our findings indicate that the teams of Cases 1, 4, 5, and 6 found the Spotify model helpful. *“During the transition phase, the Spotify model gave us orientation (4.1).”* This model presented guidelines for the cases for

**Table 4. Findings for relational mechanisms**

	<b>Business IT interaction</b>	<b>Communication and shared knowledge</b>
<b>Team 1</b>	The product owner is the contact person for the customer.	Parts of the Spotify communication model are used in the IT function. Lightning talks, presentation and other communication forms are used to share knowledge within the IT function. Scrum meetings are implemented.
<b>Team 2</b>	Team members engage in strong collaboration with the product owners who are located on the business side.	Tech-talks and other presentations to share knowledge within the IT function. Implementation of Kanban meetings.
<b>Team 3</b>	The product owner is the contact person for the business.	Implementation of Scrum meetings. The team members regularly share expert knowledge with other employees with similar interests (e.g. interest groups) within the company only if it is needed.
<b>Team 4</b>	The product manager is the contact person for the customer.	Implementation of Kanban meetings. Parts of the Spotify communication model are used in the IT function. Lightning talks, presentation and other communication forms are used to share knowledge within the IT function.
<b>Team 5</b>	The product owner is the contact person for the business.	Implementation of Scrum meetings. Parts of the Spotify communication model are used in the IT function. Other meetings for sharing knowledge are implemented.
<b>Team 6</b>	The product owner is the contact person for the customer.	Communities of practice are implemented for sharing knowledge. The communication model is similar to the Spotify model. Several presentations and talks are used to share knowledge within the IT function.

how cross-functional teams can be organized, and the cases we examined implemented similar communication and knowledge sharing environments.

Finally, we asked the interviewees to explain which key factors they see in the integration of DevOps teams to leverage competitive advantage. For example, the interviewees highlighted an improvement of innovativeness, time to market/responsiveness, agility, scalability, employee sense of responsibility, software quality, and flexibility. The CTO of Case 4 mentioned that their established DevOps teams deliver great innovation power and the fastest speed for software delivery. Interviewee 5.2 mentioned that they have a greater scalability if they have to cover performance peaks. Executive 1.1 highlighted that DevOps gives them the possibility to work really agile. Furthermore, Team Lead 3.1 said that the team members gain a higher sense of responsibility, which develops a better awareness of the service. Additionally, Executive 1.2 mentioned that the software quality is much better, because the team has a much broader knowledge of all necessary tasks in the software

development process. Team Lead 2.1 described the way they reduce waiting times and stated that they are more flexible since they assume all the roles and responsibilities within the teams.

## 5. Discussion

The goal of this study was to provide evidence of the ways that incumbent firms can implement IT governance mechanisms for DevOps team orientation. Our findings extend existing knowledge about IT governance mechanisms in DevOps teams through an exploratory qualitative research study. Past literature focused primarily on traditional IT functions and their IT governance mechanisms—e.g. De Haes and Van Grembergen [25] or Willcocks, Feeny and Olson [39]. However, there is no research available that provides IT governance mechanisms for DevOps oriented IT functions. Only a few investigations give insights into agile IT environments [9] and their transformation towards an agile working mode [21]. Thus, this research improves on our limited theoretical understanding of the DevOps phenomenon [11].

The results indicate that, when it comes to the IT governance mechanism of structure, the members of DevOps teams should be able to adapt all the tasks of a given software delivery cycle as far as possible. Therefore, there is a need for T-shaped employees in DevOps teams [40]. Prior literature shows that T-shaped persons are entrepreneurially minded persons who are able to find new opportunities, and that they learn fast from failures [41]. It is a great challenge for incumbent firms to achieve acceptance for the new collaboration culture of the DevOps. Hence, our findings indicate that the employment of an agile coach is very helpful for the transformation towards DevOps. For example, a Scrum Master guarantees that the team members follow the Scrum processes [42] and acts as their coach as well promoting the importance of collaboration within the team [43].

For generating a DevOps-oriented structure, the cases examined used different approaches. Our findings present various forms of the DevOps implementation, e.g. the foundation of a new spin-off, ad-hoc or long-term change-over from silo structures to cross-functional structures. Three companies we investigated (Cases 2, 4, and 5) organize their IT functions in hybrid fashion, providing services through traditional organized services as well as through DevOps teams. The remaining cases already have a completely decentralized structure or they are in the transition phase towards decentralization. Within a decentralized IT function, the teams have great decision-making autonomy, according to which all team members report to the IT units' heads [44]. All of the participants interviewed demonstrate that it is important to persuade the employees of the cultural aspect of DevOps. That means the employees need awareness of cultural changes in their daily mode of operation. Therefore, an ad hoc transformation can only be recommended if the employees are convinced of the value of the cultural changes. The findings of this research indicate that competitive advantage through DevOps teams can be achieved through the implementation of a decentralized or hybrid organizational structure. Cross-functional teams need decision-making autonomy to outperform traditional structures and achieve the key advantages mentioned, e.g. responsiveness and agility.

With the help of agile IT organizations, the IT function devolves to a partner instead of a service provider for the business. Thereby, the gap between business and IT functions can be reduced. Furthermore, business IT alignment is dependent on the degree of integration of a product owner within agile IT teams [9]. Our research highlighted the fact that all teams have implemented an agile software

development method and a product owner role within their DevOps teams. The interviewees emphasized that the product owner is the customer contact person, and is responsible for the requirements that should be developed. Hence, the product owner is a very important role for them and we validate the findings of Horlach, Drews, Schirmer and Boehmann [9]. In addition, our research presents the idea that communication and shared knowledge play a significant role to govern DevOps teams. By virtue of the high decentralization of DevOps-oriented IT structures, DevOps teams need highly implemented communication and knowledge-sharing opportunities. Most of the teams we examined have a collaboration environment based on the Spotify model. Hence, our study highlighted the importance of strong relational mechanisms that should be implemented within DevOps teams. Only if team members have the ability to share knowledge and communicate inside the team and with the rest of company can the DevOps culture be developed.

## 6. Implications for research and practice

Our research has implications for future research and practice. With the present research we delivered new insights into the research area of IT governance mechanisms. To be more concrete, we presented IT governance mechanisms that are important for DevOps teams. Our findings present precise IT governance mechanisms in the area of structure, processes, and relational mechanisms for DevOps teams. Hence, our contribution is that IT governance mechanisms are important for DevOps teams. In addition we confirmed that cross-functional IT teams need a lightweight governance, as mentioned in prior literature [9].

Prior research claims that cross-functional teams and traditional silo-oriented IT approaches can coexist [45]. But we found no research that provides IT governance mechanisms for the case of traditional and DevOps-oriented IT functions. Hence, with this research, IT managers gain detailed insights into the IT governance mechanisms of DevOps. We also depict how these mechanisms can be integrated within IT functions to achieve a DevOps orientation with suitable governance. Beyond that, our findings promote the idea that cross-functional collaboration can lead to competitive advantages. CIOs and IT managers should use these IT governance mechanisms to achieve the advantages offered by DevOps orientation. Finally, we present insights on how the DevOps approach can be integrated into existing organizational structures.



## 7. Limitations

This research presents insights into how incumbent firms can implement suitable IT governance mechanisms for the implementation of DevOps orientation. Some limitations should, however, be considered while interpreting the results. The generalizability of the findings is limited, because we conducted a qualitative study. We examined six different organizations but focused on only one team. Furthermore, the several cases are at different stages of the DevOps integration. Some of the teams we looked at had had a DevOps orientation for six months, others for several years. In addition, the interviews took place only in German organizations and the case study has limited participants. Further research could enhance the study in different countries, and examine more teams per organization with help of different qualitative techniques. Furthermore, in this study only interviews with IT persons were conducted. The study could be enhanced through interviews with the business side—since IT governance impacts business—and by investigating the impact on business IT alignment of DevOps teams.

## 8. Conclusion

The DevOps phenomenon is much talked about. Little is known about this approach and how incumbent companies can develop the necessary IT governance mechanisms. The findings in this study deliver insights into the implementation of these mechanisms. As past literature emphasized, agile and cross-functional team-based working environments need lightweight governance [9]. This paper provides a starting point for researchers and people in practice on how governance structure, processes, and relational mechanisms can be developed in practice with a focus on cross-functional teams. The IT governance mechanisms were derived with the help of case studies which we conducted in six different organizations. They demonstrate key governance mechanisms, for example:

- The team is able to take over all tasks of the software delivery lifecycle.
- The team has great autonomy regarding decision-making process.
- A product owner is implemented for business IT interaction within the team.
- A communication model is used for knowledge sharing and team learning.

We have extended the existing literature on IT governance mechanisms and have delivered concise

mechanisms for the governance of DevOps teams. IT managers can benefit from this guidance if they want to implement cross-functional approaches or are already in the transition phase.

## 9. Acknowledgment

This research received funding from the German Federal Ministry of Education and Research (BMBF) under grant code 03FH005PX4. The responsibility for the content of this publication lies with the author.

## 10. References

- [1] Hess, T., Matt, C., Benlian, A., and Wiesböck, F., "Options for Formulating a Digital Transformation Strategy", *MIS Quarterly Executive*, 15(2), 2016, pp. 123-139.
- [2] Leidner, D.E., Urbach, N., Drews, P., and Ross, J., "Digital Business Transformation and the Changing Role of the IT Function", *MIS Quarterly Executive*, 16(2), 2017, pp. ii-iv.
- [3] Dybå, T., and Dingsøyr, T., "Empirical Studies of Agile Software Development: A Systematic Review", *Information and Software Technology*, 50(9), 2008, pp. 833-859.
- [4] West, D., Grant, T., Gerush, M., and D'silva, D., "Agile Development: Mainstream Adoption Has Changed Agility", *Forrester Research*, 2(1), 2010, pp. 1-21.
- [5] Tripp, J.F., Riemenschneider, C., and Thatcher, J.B., "Job Satisfaction in Agile Development Teams: Agile Development as Work Redesign", *Journal of the Association for Information Systems*, 17(4), 2016, pp. 267 – 307.
- [6] Fink, L., and Neumann, S., "Gaining Agility through IT Personnel Capabilities: The Mediating Role of IT Infrastructure Capabilities", *Journal of the Association for Information Systems*, 8(8), 2007, pp. 440-462.
- [7] Coyle, S., Conboy, K., and Acton, T., "An Exploration of the Relationship between Contribution Behaviours and the Decision Making Process in Agile Teams", *International Conference on Information Systems*, 2015.
- [8] Minnesota, A.I., "Renewing the IT Curriculum: Responding to Agile, Devops, and Digital Transformation" Minnesota, A.I., 'ed.' St. Paul, MN, 2016.
- [9] Horlach, B., Drews, P., Schirmer, I., and Boehmann, T., "Increasing the Agility of IT Delivery: Five Types of Bimodal IT Organization", *Hawaii International Conference on System Sciences*, 2017.
- [10] Hedman, J., Sarker, S., and Veit, D., "Editorial Digitization in Business Models and Entrepreneurship", *Information Systems Journal*, 2016.
- [11] Fitzgerald, B., and Stol, K.-J., "Continuous Software Engineering: A Roadmap and Agenda", *Journal of Systems and Software*, (123), 2015, pp. 176–189.
- [12] Mullaguru, S.N., "Changing Scenario of Testing Paradigms Using DevOps—A Comparative Study with Classical Models", *Global Journal of Computer Science and Technology*, 15(2), 2015.

- [13] Ramesh, B., Mohan, K., and Cao, L., "Ambidexterity in Agile Distributed Development: An Empirical Investigation", *Information Systems Research*, 23(2), 2012, pp. 323-339.
- [14] Humble, J., and Farley, D., *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*, Pearson Education, Inc., Boston, MA, 2011.
- [15] Lwakatare, L.E., Karvonen, T., Sauvola, T., Kuvaja, P., Olsson, H.H., Bosch, J., and Oivo, M., "Towards DevOps in the Embedded Systems Domain: Why Is It So Hard?", *Hawaii International Conference on System Sciences* 2016.
- [16] Prasad, A., Green, P., and Heales, J., "IT Governance in Collaborative Organizational Structures", *Americas Conference on Information Systems*, 2011.
- [17] Humble, J., and Molesky, J., "Why Enterprises Must Adopt DevOps to Enable Continuous Delivery", *Cutter IT Journal*, 24(8), 2011, pp. 6.
- [18] Fitzgerald, B., and Stol, K.-J., "Continuous Software Engineering and Beyond: Trends and Challenges", *International Workshop on Rapid Continuous Software Engineering*, 2014, pp. 1-9.
- [19] Weill, P., and Ross, J.W., *IT Governance: How Top Performers Manage IT Decision Rights for Superior Results*, Harvard Business Press, Boston, USA, 2004.
- [20] Lunardi, G.L., Maçada, A.C.G., Becker, J.L., and Van Grembergen, W., "Antecedents of IT Governance Effectiveness: An Empirical Examination in Brazilian Firms", *Journal of Information Systems*, 31(1), 2016, pp. 41-57.
- [21] Hekkala, R., Stein, M.-K., Rossi, M., and Smolander, K., "Challenges in Transitioning to an Agile Way of Working", *Hawaii International Conference on System Sciences*, 2017.
- [22] Huang, R., Zmud, R.W., and Price, R.L., "Influencing the Effectiveness of IT Governance Practices through Steering Committees and Communication Policies", *European Journal of Information Systems*, 19(3), 2010, pp. 288-302.
- [23] Peterson, R., "Crafting Information Technology Governance", *Information Systems Management*, 21(4), 2004, pp. 7-22.
- [24] Van Grembergen, W., *Strategies for Information Technology Governance*, IGI Global, Hershey, 2004.
- [25] De Haes, S., and Van Grembergen, W., "IT Governance Structures, Processes and Relational Mechanisms: Achieving IT/Business Alignment in a Major Belgian Financial Group", *Hawaii International Conference on System Sciences*, 2005.
- [26] Henderson, J.C., and Venkatraman, N., "Strategic Alignment: Leveraging Information Technology for Transforming Organizations", *IBM Systems Journal*, 32(1), 1993, pp. 4-16.
- [27] Wu, S.P.-J., Straub, D.W., and Liang, T.-P., "How Information Technology Governance Mechanisms and Strategic Alignment Influence Organizational Performance: Insights from a Matched Survey of Business and IT Managers", *MIS Quarterly*, 32(2), 2014, pp. 497-518.
- [28] De Haes, S., and Van Grembergen, W., "IT Governance and Its Mechanisms", *Information Systems Control Journal*, (1), 2004, pp. 27-33.
- [29] De Haes, S., and Van Grembergen, W., "An Exploratory Study into IT Governance Implementations and Its Impact on Business/IT Alignment", *Information Systems Management*, 26(2), 2009, pp. 123-137.
- [30] Kniberg, H., and Ivarsson, A., "Spotify Engineering Culture" <https://labs.spotify.com/2014/03/27/spotify-engineering-culture-part-1/>, 2012, accessed 10.05.2017.
- [31] Yin, R.K., *Case Study Research: Design and Methods*, SAGE, Thousand Oaks, CA, 2009.
- [32] Bhattacherjee, A., *Social Science Research: Principles, Methods, and Practices*, Textbooks Collection, 2012.
- [33] Flyvbjerg, B., "Five Misunderstandings About Case-Study Research", *Qualitative Inquiry*, 12(2), 2006, pp. 219-245.
- [34] Eisenhardt, K.M., "Building Theories from Case Study Research", *Academy of Management Review*, 14(4), 1989, pp. 532-550.
- [35] Miles, M.B., and Huberman, A.M., *Qualitative Data Analysis: An Expanded Sourcebook*, Sage Publications, Thousand Oaks, 1994.
- [36] Yin, R.K., *Case Study Research: Design and Methods*, Sage Publications Inc., Thousand Oaks, CA, 2009.
- [37] Maxwell, J.A., *Qualitative Research Design: An Interactive Approach*, Sage Publications, 3rd Edition edn, Thousand Oaks, CA, 2013.
- [38] Kude, T., Bick, S., Schmidt, C., and Heinzl, A., "Adaptation Patterns in Agile Information Systems Development Teams", *European Conference on Information Systems*, 2014.
- [39] Willcocks, L., Feeny, D., and Olson, N., "Implementing Core IS Capabilities: Feeny-Willcocks IT Governance and Management Framework Revisited", *European Management Journal*, 24(1), 2006, pp. 28-37.
- [40] Hansen, M.T., and Von Oetinger, B., "Introducing T-Shaped Managers. Knowledge Management's Next Generation", *Harvard Business Review*, 79(3), 2001, pp. 106-165.
- [41] Demirkan, H., and Spohrer, J., "T-Shaped Innovators: Identifying the Right Talent to Support Service Innovation", *Research-Technology Management*, 58(5), 2015, pp. 12-15.
- [42] Baumgart, R., Hummel, M., and Holten, R., "Personality Traits of Scrum Roles in Agile Software Development Teams- a Qualitative Analysis", *European Conference on Information Systems*, 2015.
- [43] Overhage, S., and Schlauderer, S., "How Sustainable Are Agile Methodologies? Acceptance Factors and Developer Perceptions in Scrum Projects", *European Conference on Information Systems*, 2012.
- [44] Walter, S.M., Böhm, T., and Krcmar, H., "Industrialisierung Der IT – Grundlagen, Merkmale und Ausprägungen eines Trends", *HMD-Praxis der Wirtschaftsinformatik*, 44(256), 2007, pp. 6-16.
- [45] Vinekar, V., Slinkman, C.W., and Nerur, S., "Can Agile and Traditional Systems Development Approaches Coexist? An Ambidextrous View", *Information Systems Management*, 23(3), 2006, pp. 31-42.